This listing of claims will replace all prior versions, and listings, of claims in the application.

Listing of Claims:

1.   (Currently Amended) A method of storing a state store having state information therein on a computing device, the method comprising:

obtaining information at least nearly unique to the computing device, the obtained information being generally non-changing and consistently obtainable and being selected from a group consisting of a hardware identification (HWID) generated for the computing device based on one or more identifications of hardware of the computing device, and a specific time associated with the computing device;

determining a number x of locations at which at least a portion of the state store is to be stored at;

generating x pseudo-random file names and x corresponding paths based at least in part on the obtained information by applying a one-way function to data including the obtained information and employing a resulting output of the function to define the file named and the paths, whereby the generated file names and corresponding paths are likewise at least nearly unique to the computing device;

pairing the x generated file names and the x generated paths to form the x locations; and

storing the state store according to the x generated locations.

2.   (Canceled)

3.   (Original)   The method of claim 1 comprising obtaining information specific to the computing device comprising an install time of an operating system thereof.

4.   (Original)   The method of claim 1 comprising obtaining information specific to a current period of time, whereby the state store is stored according to a location that varies according to such current period of time

5.   (Original)   The method of claim 1 comprising determining the number x of locations as a number n of parts in which the state store is to be divided times a number m of copies of each part that are to be stored.

6.   (Original)   The method of claim 1 comprising generating x pseudo-random file names, each having a pseudo-random name length.

7.   (Original)   The method of claim 1 comprising generating x paths, each path comprising one of a plurality of levels of an operating system directory path on the computing device.

8.   (Original)   The method of claim 1 comprising generating x paths, each path comprising one of a plurality of levels of a registry path on the computing device.

9.   (Original)   The method of claim 1 wherein storing the state store according to the x generated locations comprises:

    protecting the state store by performing at least one of:

    signing the state store to produce a signature and appending the signature to the state store; and

    encrypting the state store to produce an encrypted state store;

    dividing the state store into n parts;

    saving each of the n parts m times according to the x = n times m formed locations.

10.   (Original)   The method of claim 1 wherein storing the state store according to the x generated locations comprises:

    dividing the state store into n parts;

    protecting the state store by signing at least one of the n parts of the state store to produce a signature and appending the signature to the part; and

    saving each of the n parts according to the x formed locations.

11.   (Original)   The method of claim 1 further comprising retrieving the stored state store, the retrieving comprising:

obtaining the information at least nearly unique to the computing device;

determining the number x of locations at which at least a portion of the state store is stored at;

generating the x pseudo-random file names and the x corresponding paths based at least in part on the obtained information;

pairing the x generated file names and the x generated paths to form the x locations; and

retrieving the state store from the x generated locations.

12.   (Original)   The method of claim 11 wherein the state store has been divided into n parts and each of the n parts has been saved according to the x formed locations, and wherein retrieving the stored state further comprises:

retrieving the n parts from the x locations;

reconstituting the state store from the retrieved n parts thereof;

if the reconstituted state store is encrypted, decrypting same; and

if the reconstituted state store is signed to produce a signature, verifying the signature.

13.   (Original)   The method of claim 12 wherein the state store has been divided into n parts and each of the n parts has been saved m times according to the x formed locations, and wherein retrieving the stored state comprises reconstituting m copies of the state store from the retrieved n parts thereof, and further comprises randomly selecting one of the m reconstituted copies.

14.   (Original)   The method of claim 1 wherein each file name has a length and wherein generating x pseudo-random file names based at least in part on the obtained information comprises:

hashing data including the obtained information to produce a first hash comprising a string of numbers;

for each file name length, applying a pre-defined serial portion of the first hash to a function to result in the file name length; and

for each Nth file name:

performing a predetermined modification to the Nth hash;

hashing the modified Nth hash to produce an (N+1)th hash comprising a string of numbers, whereby the first hash is employed to produce a second hash for the first file name, the second hash is employed to produce a third hash for the second name, etc.; and

for each file name character of the Nth file name, applying a pre-defined serial portion of the (N+1)th hash to a function to result in the file name character.

15.   (Original)   The method of claim 14 wherein each file name length has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the first hash to a function to result in the file name length comprises applying the pre-defined serial portion of the first hash to the modulo function:

Length = [serial portion mod (maximum – minimum)] +minimum.

16.   (Original)   The method of claim 14 wherein applying the pre-defined serial portion of the (N+1)th hash to a function to result in the file name character comprises applying the pre-defined serial portion of the first hash to a conversion table predefined for the computing device.

17.   (Original)   The method of claim 14 wherein the modification comprises at least one of a bit shift, a reverse ordering, and a swapping.

18.   (Original)   The method of claim 1 wherein each path comprises one of a plurality of levels of an operating system directory path on the computing device, and wherein generating x paths based at least in part on the obtained information comprises:

hashing data including or based on the obtained information to produce a path hash comprising a string of numbers;

for each path, applying a pre-defined serial portion of the path hash to a function to result in a level for the path.

19.   (Original)   The method of claim 18 wherein each path level has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the path hash to a function to result in the level for the path comprises applying the pre-defined serial portion of the path hash to the modulo function:

Level = [serial portion value mod (maximum – minimum)] +minimum.

20.   (Original)   The method of claim 1 comprising defining successive periods of time, and for each successive period of time:

obtaining information at least nearly unique to the computing device;

determining a number x of locations at which at least a portion of the state store is to be stored at;

generating x pseudo-random file names and x corresponding paths based at least in part on the obtained information and based at least in part on indicia relevant to the period of time, whereby the generated file names and corresponding paths are likewise at least nearly unique to the computing device and unique to the period of time;

pairing the x generated file names and the x generated paths to form the x locations; and

storing the state store according to the x generated locations, whereby the state store is moved during each successive period of time.

21.   (Original)   The method of claim 1 further comprising:

obtaining alternate information relevant to the computing device;

generating at least one pseudo-random file name and at least one corresponding path based at least in part on the alternate information and pairing same to form at least one alternate location; and

storing the obtained information as original information according to the at least one generated alternate location, whereby if the obtained information changes on the computing

device, such changed information cannot be employed to retrieve the state store but the alternate information can be employed to retrieve the original information and the original information can be employed to retrieve the state store.

22.   (Currently Amended) A computer-readable medium having stored thereon computer-executable instructions for performing a method of storing a state store having state information therein on a computing device, the method comprising:

obtaining information at least nearly unique to the computing device, the obtained information being generally non-changing and consistently obtainable and being selected from a group consisting of a hardware identification (HWID) generated for the computing device based on one or more identifications of hardware of the computing device, and a specific time associated with the computing device;

determining a number x of locations at which at least a portion of the state store is to be stored at;

generating x pseudo-random file names and x corresponding paths based at least in part on the obtained information by applying a one-way function to data including the obtained information and employing a resulting output of the function to define the file named and the paths, whereby the generated file names and corresponding paths are likewise at least nearly unique to the computing device;

pairing the x generated file names and the x generated paths to form the x locations; and

storing the state store according to the x generated locations.

23.   (Canceled)

24.   (Original)   The medium of claim 22 wherein the method comprises obtaining information specific to the computing device comprising an install time of an operating system thereof.

25.   (Original)   The medium of claim 22 wherein the method comprises obtaining information specific to a current period of time, whereby the state store is stored according to a location that varies according to such current period of time

26.   (Original)   The medium of claim 22 wherein the method comprises determining the number x of locations as a number n of parts in which the state store is to be divided times a number m of copies of each part that are to be stored.

27.   (Original)   The medium of claim 22 wherein the method comprises generating x pseudo-random file names, each having a pseudo-random name length.

28.   (Original)   The medium of claim 22 wherein the method comprises generating x paths, each path comprising one of a plurality of levels of an operating system directory path on the computing device.

29.   (Original)   The medium of claim 22 wherein the method comprises generating x paths, each path comprising one of a plurality of levels of a registry path on the computing device.

30.   (Original)   The medium of claim 22 wherein storing the state store according to the x generated locations comprises:
       protecting the state store by performing at least one of:
       signing the state store to produce a signature and appending the signature to the state store; and
       encrypting the state store to produce an encrypted state store;
       dividing the state store into n parts;
       saving each of the n parts m times according to the $x = n$ times m formed locations.

31.   (Original)   The medium of claim 22 wherein storing the state store according to the x generated locations comprises:
       dividing the state store into n parts;

protecting the state store by signing at least one of the n parts of the state store to produce a signature and appending the signature to the part; and

saving each of the n parts according to the x formed locations.

32.   (Original)   The medium of claim 22 wherein the method further comprises retrieving the stored state store, the retrieving comprising:

obtaining the information at least nearly unique to the computing device;

determining the number x of locations at which at least a portion of the state store is stored at;

generating the x pseudo-random file names and the x corresponding paths based at least in part on the obtained information;

pairing the x generated file names and the x generated paths to form the x locations; and

retrieving the state store from the x generated locations.

33.   (Original)   The medium of claim 32 wherein the state store has been divided into n parts and each of the n parts has been saved according to the x formed locations, and wherein retrieving the stored state further comprises:

retrieving the n parts from the x locations;

reconstituting the state store from the retrieved n parts thereof;

if the reconstituted state store is encrypted, decrypting same; and

if the reconstituted state store is signed to produce a signature, verifying the signature.

34.   (Original)   The medium of claim 32 wherein the state store has been divided into n parts and each of the n parts has been saved m times according to the x formed locations, and wherein retrieving the stored state comprises reconstituting m copies of the state store from the retrieved n parts thereof, and further comprises randomly selecting one of the m reconstituted copies.

35.   (Original)   The medium of claim 22 wherein each file name has a length
and wherein generating x pseudo-random file names based at least in part on the obtained
information comprises:

hashing data including the obtained information to produce a first hash comprising a
string of numbers;

for each file name length, applying a pre-defined serial portion of the first hash to a
function to result in the file name length; and

for each Nth file name:

performing a predetermined modification to the Nth hash;

hashing the modified Nth hash to produce an (N+1)th hash comprising a string of
numbers, whereby the first hash is employed to produce a second hash for the first file name,
the second hash is employed to produce a third hash for the second name, etc.; and

for each file name character of the Nth file name, applying a pre-defined serial portion
of the (N+1)th hash to a function to result in the file name character.

36.   (Original)   The medium of claim 35 wherein each file name length has a
preset minimum and maximum, and wherein applying the pre-defined serial portion of the
first hash to a function to result in the file name length comprises applying the pre-defined
serial portion of the first hash to the modulo function:

Length = [serial portion mod (maximum − minimum)] +minimum.

37.   (Original)   The medium of claim 35 wherein applying the pre-defined
serial portion of the (N+1)th hash to a function to result in the file name character comprises
applying the pre-defined serial portion of the first hash to a conversion table predefined for
the computing device.

38.   (Original)   The medium of claim 35 wherein the modification comprises at
least one of a bit shift, a reverse ordering, and a swapping.

39.   (Original)   The medium of claim 22 wherein each path comprises one of a plurality of levels of an operating system directory path on the computing device, and wherein generating x paths based at least in part on the obtained information comprises:

hashing data including or based on the obtained information to produce a path hash comprising a string of numbers;

for each path, applying a pre-defined serial portion of the path hash to a function to result in a level for the path.

40.   (Original)   The medium of claim 39 wherein each path level has a preset minimum and maximum, and wherein applying the pre-defined serial portion of the path hash to a function to result in the level for the path comprises applying the pre-defined serial portion of the path hash to the modulo function:

Level = [serial portion value mod (maximum − minimum)] +minimum.

41.   (Original)   The medium of claim 22 wherein the method comprises defining successive periods of time, and for each successive period of time:

obtaining information at least nearly unique to the computing device;

determining a number x of locations at which at least a portion of the state store is to be stored at;

generating x pseudo-random file names and x corresponding paths based at least in part on the obtained information and based at least in part on indicia relevant to the period of time, whereby the generated file names and corresponding paths are likewise at least nearly unique to the computing device and unique to the period of time;

pairing the x generated file names and the x generated paths to form the x locations; and

storing the state store according to the x generated locations, whereby the state store is moved during each successive period of time.

42.   (Original)   The medium of claim 22 wherein the method further

comprises:

obtaining alternate information relevant to the computing device;

generating at least one pseudo-random file name and at least one corresponding path

based at least in part on the alternate information and pairing same to form at least one

alternate location; and

storing the obtained information as original information according to the at least one

generated alternate location, whereby if the obtained information changes on the computing

device, such changed information cannot be employed to retrieve the state store but the

alternate information can be employed to retrieve the original information and the original

information can be employed to retrieve the state store.